REMARKS

The Examiner is thanked for the performance of a thorough search. By this amendment Claims 5, 10, 12, 16, 17, 23, 28, 30 and 34-35 have been amended. No claims have been cancelled or added. Hence, Claims 5-14, 16-18, 23-32 and 34-36 are pending in the application.

SUMMARY OF THE REJECTIONS

Claims 5-15, 16, 18, 23-32 and 34-36 have been rejected under 35 U.S.C. § 101 for allegedly being directed to non-statutory subject matter.

Claims 5, 12-14 and 28-32, and 35 have been rejected under 35 U.S.C. § 112, second paragraph, as allegedly being indefinite.

Claims 12-14 and 30-32 have been rejected under 35 U.S.C. § 102(b) as allegedly unpatentable over an article entitled "Recovering in Large Distributed Systems with Replicated Data" by Peter Triantafillou (hereinafter referred to as "*Peter*").

Claims 10-11 and 28-29 have been rejected under 35 U.S.C. § 102(e) as allegedly unpatentable over U.S. Patent No. 5,966,706 issued to Biliris et al. (hereinafter referred to as "*Biliris*").

The rejections are respectfully traversed.

THE PENDING CLAIMS CONFORM TO 35 U.S.C. § 101

Claim 5-15, 16, 18, 23-32 and 34-36 were rejected under 35 U.S.C. § 101 as allegedly being directed to non-statutory subject matter. The Office Action alleges that the subject matter featured in each of Claims 5-15, 16, 18, 23-32 and 34-36 represents an abstract idea without featuring a claimed limitation drawn to a practical application.

Without acquiescence to the position of the Office Action or prejudice to pursue the claims as originally filed in a continuation application, each of Claims 5, 10, 12, 16, 17, 23, 28, 30 and 34-35 has been amended to clarify that the claim is directed towards an approach for recovering from a failure of one or more nodes of a multiple-node system. Recovering from a failure of one or more nodes of a multiple-node system is tied to a technology art and has a practical application that produces a concrete, useful, and tangible result. As a result, each of Claims 5-15, 16, 18, 23-32 and 34-36 form the basis of statutory subject matter under 35 U.S.C. § 101.

Independent Claims 5, 10, 12, 16, and 17 have also been amended to clarify that the claim is directed to a computer executable method in accordance with the suggestion of the Office Action. The computer executable methods of Claims 5, 10, 12, 16, and 17 each feature at least one element that is performed by the execution of a computer. Therefore, the rejection of Claims 5-14 and 16-18 under 35 U.S.C. § 101 for allegedly being directed to an abstract idea is overcome.

THE PENDING CLAIMS CONFORM TO 35 U.S.C. § 112, SECOND PARAGRAPH

Claims 12-14 and 28-32 have been rejected under 35 U.S.C. § 112, second paragraph, on grounds that "a method for recovering after a failure", as recited in the preamble, is allegedly vague and indefinite. The Office Action alleges:

> it is unclear what is being recovered after a failure. Without providing any indication, one having ordinary skill in the art would not know what has been recovered

As stated above, each of Claims 5-15, 16, 18, 23-32 and 34-36 has been amended to clarify that the claim is directed towards an approach for recovering from a failure of

one or more nodes of a multiple-node system. Additionally, the language of the preamble, when viewed with the benefit of the Applicants' specification, is clear in both meaning and scope (see paragraphs 132-135, under the heading entitled "Recovery"). Consequently, it is respectfully submitted that the rejection of Claims 12-14 and 30-32 under 35 U.S.C. § 112, second paragraph has been overcome.

Additionally, Claims 5, 32 and 35 were rejected under 35 U.S.C § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The Office Action stated that Claims 5, 32 and 35 were rejected for the use of the pronoun "that." Claims 5, 32 and 35 have been amended to cease the recitation of the offending pronoun. Consequently, it is respectfully submitted that amended Claims 5, 32 and 35 overcome the objections of the Office Action under 35 U.S.C § 112, second paragraph.

THE PENDING CLAIMS ARE PATENTABLE OVER THE CITED ART

Each of the pending claims recites a combination of features that are not disclosed, taught, or suggested by the cited art, either individually or in combination. Each pending claim shall be discussed below.

Claim 12

Claim 12 recites the features of:

> A computer executable method for recovering after a failure of one or more nodes of a multiple-node system, the method comprising the steps of:
> if it is unclear whether a particular version of a data item has been written to disk, then performing the steps of
> **without attempting to recover said data item, marking dirtied cached versions of said data item that would have been covered if said particular version was written to disk;**

when a request is made to write one of said dirtied cached versions to disk, determining which version of said data item is already on disk; and
if said particular version of said data item is already on disk, then not writing said one of said dirtied cached versions to disk. (emphasis added)

At least the above-bolded elements are not disclosed, taught, or suggested by *Peter*.

While Claim 12 and *Peter* are both directed towards recovery of a system, beyond this broad generality there are fundamental differences between the approach of Claim and that of *Peter*. Claim 12 is directed towards recovery after a failure of one or more nodes of a multi-node system. A determination is made as to whether it is unclear that a particular version of a data item has been written to the disk. If it is unclear whether the particular data item was written to disk, then without attempting to recover the data item, a dirtied cached version of the data item that would have been covered if the particular version were written to disk is marked. A "dirty data item" is a data item that (1) contains changes, and (2) has not yet been persistently stored (Specification Page 3 paragraph 8). When a request is made to write one of the dirtied cached versions of the data item to disk, a determination is made as to which version of the data item is already on disk. If the particular version of the data item is already on disk, then the dirtied cached version of the data item is not written to disk.

On the other hand, *Peter* is directed towards recovery in large-scale transaction-based distributed systems with replicated data. *Peter* discusses an approach for recovering from a crash by examining a log file that may contain an uncertain flag. An uncertain flag indicates that data may be, but is not necessarily, out of date (page 40, fifth paragraph). However, the log file of *Peter* is stored on disk. *Peter* lacks any suggestion

of using a cache to recover from a crash. As a result, *Peter* lacks any suggestion of a dirty data item.

In view of the fundamental differences between Claim 12 and *Peter*, numerous features of Claim 12 are not disclosed, taught, or suggested by *Peter*. For example, Claims 12 recites the feature of "without attempting to recover said data item, marking dirtied cached versions of said data item that would have been covered if said particular version was written to disk" featured in Claim 12. However, the portion of *Peter* cited to show this element (page 40, Col. 2) lacks any suggestion of taking any action on any data item in a cache, let alone "marking dirtied cached versions of said data item that would have been covered if said particular version was written to disk" as required by this element. At best, this portion discusses marking uncertain transaction in a log; however, the log must be stored in persistent storage in order to recover from the crash. As a result, there is nothing analogous to a dirtied cached version of a data item in the cited portion of *Peter*, as any information written in the log must be persistently stored, and therefore, cannot qualify as a dirtied cached version of a data item. Consequently, this element cannot be disclosed, taught, or suggested by *Peter*.

Further, Claim 12 recites the feature of "when a request is made to write one of said dirtied cached versions to disk, determining which version of said data item is already on disk." However, the portion of *Peter* cited to show this element (at page 40, Col. 2, items 1 and 2) lacks any suggestion of a (a) a dirtied cached version of a data item, or (b) determining which version of a data item is already on disk. As a result, the cited portion of *Peter* cannot possibly show determining which version of a data item is already on disk when a request is made to write a dirtied cached version to disk. Instead, this portion of *Peter* merely discusses two mutually exclusive ways to erase an "uncertain

flag" from a log file by either performing a write operation or consulting reading replicated data; however, neither of these ways to erase an uncertain flag from a log file even involves writing data from cache to disk. As a result, this element cannot be disclosed, taught, or suggested by *Peter*.

Moreover, Claim 12 features the element of "if said particular version of said data item is already on disk, then not writing said one of said dirtied cached versions to disk" featured in Claim 12. However, the portion of *Peter* cited to show this element (at page 40, Col. 2, items 1 and 2) merely discusses an approach for erasing an "uncertain flag" of a log file. Since the log file is stored on disk, the operations discussed in this portion of *Peter* involve operations performed on data known to be stored on disk. Thus, this portion of *Peter* lacks any suggestion of determining whether a particular version of a data item is already on disk. As a result, this portion of *Peter* cannot disclose, teach, or suggest this element.

As *Peter* fails to disclose, teach, or suggest at least one element recited in Claim 12, it is respectfully submitted that Claim 12 is patentable over the cited art, and is in condition for allowance.

Claim 10

Claim 10 recites the features of:

> A computer executable method for recovering after a failure of one or more nodes in a multiple-node system, the method comprising the steps of:
> **determining whether the failure involves only one node; and**
> **if the failure involves only said one node, then performing recovery by applying a recovery log of said node beginning at a first point in the recovery log; and**
> **if the failure involves one or more nodes in addition to said one node, then performing recovery by applying said recovery log of said node beginning at a second point in the recovery log;**

wherein said first point is different from said second point. (emphasis added)

At least the above-bolded elements are not disclosed, taught, or suggested by *Biliris*.

While Claim 10 and *Biliris* are both directed towards recovery of a system, beyond this broad generality there are fundamental differences between the approach of Claim and that of *Biliris*. Claim 10 is directed to an approach for recovering after a failure of one or more nodes. According to the approach of Claim 10, a determination is made as to whether the failure involves only one node or multiple nodes. If the failure involves only one node, then recovery is performed by applying a recovery log of the failed node beginning at a first point in the recovery log. However, if the failure involves one or more nodes in addition to the first node, then recovery is performed by applying the recovery log of the additional node beginning at a second point in the recovery log. According to the approach of Claim 10, recovery is performed by applying the same recovery log regardless of whether the failure involved one node or two or more nodes.

On the other hand, *Biliris* is directed towards performing local logging in a distributed database management computer system. In *Biliris*, each node performs local logging. As a result, in *Biliris* it is necessary for a crashed node to not only scan its own log file, but to communicate with other nodes to obtain a variety of information contained in log files of other nodes in the recovery process of the crashed node. Such an approach is fundamentally different then the features of Claim 10.

In view of the fundamental differences between Claim 10 and that of *Biliris*, numerous features of Claim 10 are not disclosed, taught, or suggested by *Biliris*. For example, Claim 1 recites the features "if the failure involves only said one node, then performing recovery by applying a recovery log of said node beginning at a first point in the recovery log." In *Biliris*, regardless of whether the failure involves a single node or

two or more nodes, recovery is not performed by applying a recovery log of the node that failed. Instead, the cited portions of *Biliris* describe performing recovery of a crashed node by the crashed node engaging in a series of communications between other nodes to obtain information about which pages need to be recovered, which nodes are involved in the recovery, reconstructing lock information, and coordinating the recovery among the involved nodes. These communications involve scanning the log files of other nodes besides the crashed node.

To illustrate, *Biliris* teaches when a crashed node restarts, "it requests from each operational node N, the list of all pages owned by N that are present in a remote node's (N_r's) cache, as well as all entries in N_r's DPT that correspond to pages owned by N" (See Col. 10; lines 41-44). Further, *Biliris* teaches "when a remote node N_r receives the list of pages that require recovery from N, it scans its log file starting from the minimum of all RedoLSN values belonging to DPT entries for the pages that are included in the above list (see Col. 11, lines 57-60). Thus, it is clear that in the approach of *Biliris*, the recovery of a failed node cannot be performed by applying a recovery log of the failed node (i.e., a single recovery log), as featured in Claim 10.

As a result, *Biliris* cannot disclose, teach, or suggest the elements of "if the failure involves only said one node, then performing recovery by applying a recovery log of said node beginning at a first point in the recovery log; and if the failure involves one or more nodes in addition to said one node, then performing recovery by applying said recovery log of said node beginning at a second point in the recovery log; wherein said first point is different from said second point" featured in Claim 10. Instead, as explain above, the approach of *Biliris* requires that the crashed node request information from each operational node.

Further, no portion of *Biliris* discloses, teaches, or suggests the element of "determining whether the failure involves only one node" featured in Claim 12. The portion of *Biliris* cited to show this element (Col. 9, lines 35-50; Col. 10, lines 8-12; Col. 10, lines 66-67; Col. 11, lines 2-14) lacks any suggestion of determining whether a failure involves only one node. Indeed, there would be no need for *Biliris* to make such a determination, since regardless of how many nodes encounter a failure, the method for recovery of each failed node in *Biliris* is the same. Thus, this element is also not disclosed, taught, or suggested by *Biliris*.

Moreover, no portion of *Biliris* discloses, teaches, or suggests:

> "if the failure involves one or more nodes in addition to said one node,
> then performing recovery by applying said recovery log of said node beginning at a second point in the recovery log;
> wherein said first point is different from said second point"

Instead, the portion of *Biliris* cited to show this feature (Col. 12, lines 60-67; Col. 13, lines 3-12) merely discusses obtaining information from a set of operational nodes used to recover the crashed node. In other words, the crashed node does not recover using information maintained at the crashed node. As a result, nothing in the cited portion of *Biliris* could be analogous to a recovery log as claimed.

As *Biliris* fails to disclose, teach, or suggest one or more elements recited in Claim 10, it is respectfully submitted that Claim 10 is patentable over the cited art, and is in condition for allowance.

Claims 13-14, 11 and 28-32

Independent Claim 28 recites features similar to those discussed above with respect to Claim 10, except that Claim 28 is recited in computer-readable medium format. Independent Claim 30 recites features similar to those discussed above with respect to

Claim 12, except that Claim 30 is recited in computer-readable medium format. Consequently, it is respectfully submitted that Claims 28 and 30 are patentable over the cited art, and are each in condition for allowance, for at least the reasons given above with respect to Claims 10, and 12 respectively.

Claims 13-14, 11, 29 and 31-32 are dependent claims, each of which depends (directly or indirectly) on one of the claims discussed above. Each of Claims 13-14, 11, 29 and 31-32 is therefore allowable for the reasons given above for the claim on which it depends. In addition, each of Claims 13-14, 11, 29 and 31-32 introduces one or more additional limitations that independently render it patentable. However, due to the fundamental differences already identified, to expedite the positive resolution of this case, a separate discussion of those limitations is not included at this time. The Applicants reserve the right to further point out the differences between the cited art and the novel features recited in the dependent claims.

CONCLUSION

For the reasons set forth above, it is respectfully submitted that all of the pending claims are now in condition for allowance. Therefore, the issuance of a formal Notice of Allowance is believed next in order, and that action is most earnestly solicited.

The Examiner is respectfully requested to contact the undersigned by telephone if it is believed that such contact would further the examination of the present application.

A petition for extension of time, to the extent necessary to make this reply timely filed, is hereby made. If any applicable fee is missing or insufficient, throughout the pendency of this application, the Commissioner is hereby authorized to charge any applicable fees and to credit any overpayments to our Deposit Account No. 50-1302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Christopher J. Brokaw
Reg. No. 45,620
Date: October 31, 2005

2055 Gateway Place, Suite 550
San Jose, CA 95110
Telephone No.: (408) 414-1080 ext. 225
Facsimile No.: (408) 414-1076

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: **Mail Stop Amendment**, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

on October 31, 2005 by Angelica Maloney